IMPROVEMENT OF NETWORK EFFICIENCY ON THE GROUNDS OF CHANGE COMMUNICATION PROTOCOL

Jiri VYCHODIL[1], *Martin MIKULEC*[1], *Jakub SAFARIK*[1], *Miroslav VOZNAK*[1]

[1]Department of Telecommunications, Faculty of Electrical Engineering and Computer Science,
VSB-Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava-Poruba, Czech Republic

jiri.vychodil@vsb.cz, martin.mikulec@vsb.cz, jakub.safarik.st1@vsb.cz, miroslav.voznak@vsb.cz

Abstract. *This article deals with both theoretical and practical application of Extensible Messaging and Presence Protocol instead of different solution described detailed in the article. It uses periodically HTTP method POST to gain information about a state, which a client stands in. During this process a new connection is created and closed, what enormously makes network traffic mote intensive. Instead of this we used XMPP, this technology eliminates disadvantages of synchronous HTTP connection by its adapting for real time connection. It uses stable connection, what does not need to create it any time any information needs to be sent. It decreased amount of transported data through a network and a time, in which is a client informed about an event due to a fact, that the new solution works in real time.*

Keywords

AJAX, asterisk, CRM, XMPP.

1. Introduction

Telecommunications and informatics are branches of science, which belong to the most evolving parts. New solutions are being discovered all the time and offer more and more complex functions. Users require multipurpose systems which will save their work and time. One of these systems is Customer relationship management (CRM), [1]. It makes communication with employees and companies easier.

The user of the system uses his computer as a thin-client to log into to web application, which is connected to a server, where all the information is stored. The client can be implemented in various program languages. Nowadays, Asynchronous JavaScript and XML (AJAX) in pair with (x)HTML or HTML client are used [2]. JavaScript serves the logic of client. Server can serve from tens to hundreds of thin-clients. The main topic, the article is focused on, is communication between client and server. Vtiger was chosen as an implementation of CRM. This product is considered to be used in a wide spectrum of companies. Internal and external relationship can be defined in the system. Suppliers, subscribers and customers can be reckoned as external. Dates of meetings, volume of orders and other data can be defined in Vtiger. There are many ranges of responsibilities defined for employees in different positions. We can assign a phone number, e-mail address, JID (Jabber ID) or another for both, internal and external users. We are mainly aimed at JID – an identifier for instant messaging. Nowadays the jabber protocol is known as XMPP [3]. Our new solution uses above mentioned BOSH, which reduces the number of HTTP requests sent to the server and network capability.

2. Present Solution

Link between Vtiger and public exchange (PBX) Asterisk is the one of main functions [4], [5]. Its user can be informed about information who is just calling him. It allows to gain information not only about the phone number, but also about all data stored in the CRM, for example a name of the calling party, name of the company and all obligations set between the companies. Quick employee's reaction can improve the company's image.

As we said before, this article deals with protocols of connection between client and server. Classic HTTP is used in original Vtiger system. This way is not very suitable, because HTTP protocol is request-response type of protocol, belonging to the application layer using TCP. The old way of CRM implementation makes TCP connection each 5 seconds to transport data. But overhead is very increased due to establishing and almost prompt connection decline, especially, when no really valuable data are transmitted. We can see the technical solution of described the situation at the Fig. 1.

Let us assume that user Bob is logged in to the Vtiger system. Alice is calling Bob, the call is going to Asterisk, then the protocol is request-response based, the

server has to wait 5 seconds to client's request, to let client know, that somebody is calling. Server hands the request by looking into the database and checking the records. In this case, HTTP response is sent and the client finally let user know who is calling. The message about a phone number of calling part and link to the website is appeared in Bob's web client.

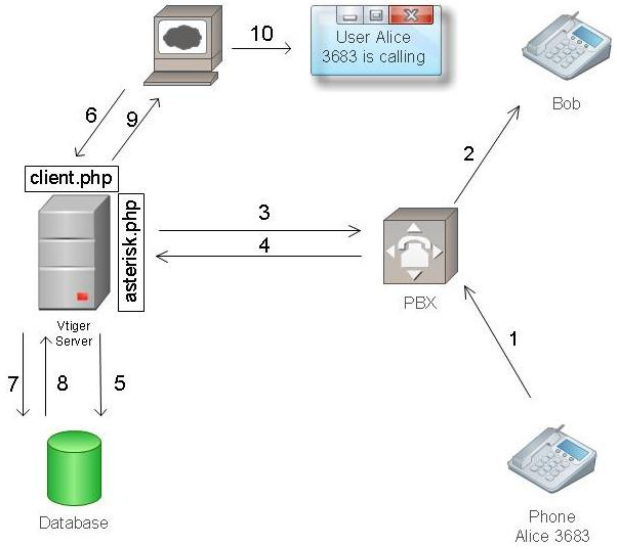

Fig. 1: Old solution using HTTP.

The main disadvantage lies in higher demands on the server and network capability. More database requests and cycling establishing and breaking-up TCP connection are also requested, which are not necessary in the optimal solution.

We have found out, that the amount of data, which are transported every five seconds take 4 kB of network capability. It seems not to be very much, but we have to consider, that there is not just one client, but even tens or hundreds. Then the network traffic is much higher. And what more, if we assume, that one hundred of clients are in use, 20 TCP connections on an average are made in a second. This is really big unnecessary amount.

3. Proposed Solution

Due to disadvantages of present solution we proposed new one. We have decided to use Extensible Messaging and Presence Protocol (XMPP). This protocol is used in communication between XMPP server and web client. Using this solution require new network elements. The situation is shown in Fig. 2.

The change is that there is Openfire – XMPP server. This software is widely spread and used. Many of companies use XMPP for their internal communication, so one more server is not considered to be a problem. The renewal issue is the way of communication between web client and Vtiger. Now the data are not transferred with the use of HTTP, but information about incoming call is transferred to the web client from Vtiger through Openfire by XMPP.

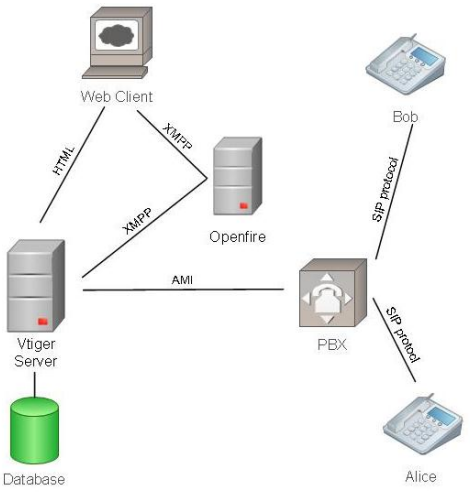

Fig. 2: Scheme of the new solution showing used protocols.

4. Used Technologies and Software

Asterisk was chosen to be used as PBX. The only one modification was made to allow communication between Vtiger and Asterisk thru AMI interface with management account.

Openfire project was chosen as XMPP server because of user-friendly environment. There are many external plugins including interconnection between Asterisk PBX and Openfire. Own XMPP server brings advantages of quick and secure communication inside the network in real operation. Openfire can be easily integrated into current information systems, because of support of LDAP or Active Directory authentication methods. Vtiger CRM, an open source CRM application uses PHP a MySQL and AJAX.

Extensible Messaging and Presence Protocol is an open-standard communications protocol for message-oriented middleware based on XML (Extensible Markup Language). The protocol was originally named Jabber and still is very often called so. Openfire was used as a server. The web Vtiger client was modified. Now it use SparkWeb - version of XMPP client which is run in web browser with the use of AJAX. Now the web client is registered and logged in the Openfire and waits for information from Vtiger server about incoming call. The server part is implemented with the use of xmpphp library. So there are four servers – Vtiger, XMPP, Asterisk and database. They can be integrated into less count of physical machines of course or the topology can be changed in many other ways. We chose a solution with two servers – Vriger with database and Openfire with PBX, because it approaches real situation in our opinion.

Many companies can use XMPP server as a tool of intercompany communication. The employees can have their XMPP account already set, so there is no need to

made new ones. The same can be used also for Vtiger thin-client, which has to be set to communicate only with Vtiger and not with other users.

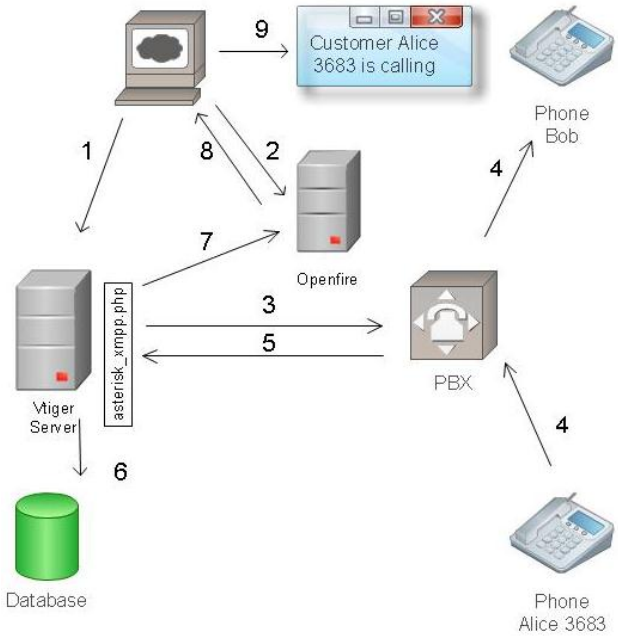

Fig. 3: Function scheme of the new solution with the use of XMPP.

Openfire uses two interfaces for communication. The classic one TCP with port 5222 for non-AJAX clients, the second one is used with TCP port 80 to communicate with clients using BOSH.

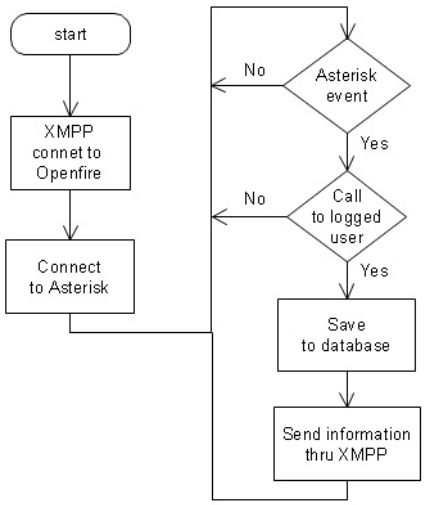

Fig. 4: Scheme of work of PHP server handling messaging.

The principle of solution is shown at Fig. 3. A user logs in to Vtiger web client and simultaneously in to XMPP server Openfire. This connection uses TCP connection. Client sends request POST, asking if there are any new incoming calls. Because there are not, XMPP server sends negative answer. Vtiger is connected with Openfire with jabber and with Asterisk with AMI interface. If Alice is calling Bob Vtiger will get message and the information will be written into database and checked, if Bob is logged into jabber server and let him know thru Openfire. Openfire resend this message to Bob thru still opened TCP connection with a link to web page containing all information about calling part.

A scheme of Vtiger function during monitoring calls in Asterisk is shown at Fig. 4. After starting, the Vtiger server is connected to Asterisk and Openfire. Then it starts watching events in Asterisk. In the case of incoming call directed to the user, whose phone number belongs to the company and the user is logged in jabber, the information about incoming call is sent to the user's web thin-client. There is one more function, which is enabled by communication between client and CRM server.

During browsing thru company information in CRM, Alice can dial a Bob's phone number. It can be done by clicking the phone number. The web client uses the XMPP to send request for call establish. The request is sent to CRM server, which saves the information for futher use to the database, then makes a request for the dial to Asterisk. It sends an Invite message to Alice's phone. After it is ringing and answering, Asterisk sends Invite message to Bob's phone. When the call is finished, CRM again writes the information into the database.

5. Measurement Achieved Results

To prove an effect of our idea we have measured network traffic. We have run a measure application tshark on a PBX server. So we can measure both communication between Vtiger and PBX and between XMPP server and client. We have set tshark to run until it catchs thousand packets on the interface. No filter was applied, so all real traffic was captured.

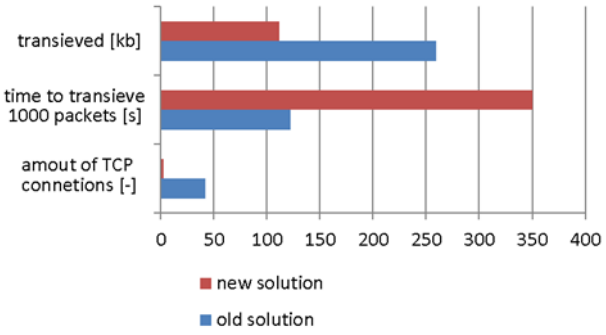

Fig. 5: Chart showing network traffic.

We can see in Fig. 5, which total number of transferred data has decreased to half. Time to transfer 1000 packets has increased almost three times. So we can assume, that small packets are now transferred due to other traffic unrelated to CRM, as DHCP, DNS, ARP and others. Amount of created TCP connections decreased from 42 to 2. These should be XMPP connection betweenweb client and XMPP server and AMI connection between server Vtiger server and Asterisk. The original solution is very limited by the time, in which

a user is informed about incoming call. This time can reach five second. It can be very long delay in real time applications. With our improvement, the delay is shorted almost to the non-measurable amount. The delay is now caused only by server handing, not by transmission assembling.

We can notice, that information about the call is transmitted twice – from CRM to XMPP server and then from XMPP server to client. But if we consider all the data uselessly transmitted in origin solution, the amount of data is still much lower. Another possible solution can be considered instead of using XMPP. The connection between web AJAX client and CRM can be realized by using WebSockets. This solution is also based on HTTP and also reduces requirements for network resources. Futhermore, WebSockets need to be implemented in web browsers, but the support is not perfect, because the specification of WebSockets is not finished yet [6].

Asterisk can handle with jabber messages after installing a module into it. We did not use this solution, because other users of XMPP would have to have Asterisk user in their contact lists. Besides, a dialplan would have to be changed, orders JabberRecieve and JabberSend functions would have to be add into each part of it. In the case of complex dialplan it would even be more chaotic. Futhermore, Asterisk does not know, which users have jabber account activated. So this solution would bring more disadvantages than advantages.

Is also possible to run XMPP server implemented in PHP. It could be done on the same machine with CRM. But the problem how the PHP client and server communicate would have to be solved.

6. Conclusion

This article deals with improving current solution of connection CRM system Vtiger with Asterisk PBX. It is based on HTTP protocol and on sending periodical requests for information to Vtiger. This extensively increases computation demands to service management and increase time delay to receive an answer by client. This solution was improved by implementation of XMPP protocol. We have managed to improve a method of sending information from client to the server in a case, when the communication is not regular. We changed the way of connection, instead of used HTTP we choose to use XMPP, which is based on TCP transfers.

In consequence, the information of incoming call is delivered to the user almost immediately. The time depends only on the processing of the information in servers and network devices. Although the information is transmitted twice – between CRM and XMPP server and then to XMPP client, the total amount of data is very decreased. Although the solution can be realized in one server, standalone server was installed for each of used the service, to approach the solution in real practice.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

References

[1] PIEPIORRA, Frank. Vtiger CRM v5.2.x: Users and Administration Manual. In: *Vtiger* [online]. 2011. Available: https://www.vtiger.com/books/vtiger-crm-users-and-administration-manual-for-v52x/.

[2] BRINZAREA, B. *AJAX and PHP: Building Responsive Web Applications*. 1st ed. Birmingham: Packt Publishing, 2006. ISBN 978-1-904811-82-4.

[3] MOFFIT, John. *Professional XMPP Programming with JavaScript and jQuery*. London: Wrox, 2010. ISBN 978-0470540718.

[4] MEGGELEN, Jim van, J. SMITH and L. MADSEN. *Asterisk The Future of Telephony*. 2nd ed. Sebastopol: O'Reilly Media, 2007. ISBN 978-0-596-00962-5.

[5] VYCHODIL, J., K. TOMALA and M. VOZNAK. Utilization Asterisk for Service Desk. *Advances Electrical And Electronic Engeneering.* 2010, vol. 8, no. 3, pp. 54-57. ISSN 1336-1376.

[6] VYCHODIL, J., F. REZAC, K. TOMALA and M. VOZNAK. Implementing WebSockets in Web applicaions. In: *Knowledge in Telecommunication Technologies and Optics 2011 - KTTO 2011*. Szczyrk: VSB-Technical University of Ostrava, 2011, pp. 136-139. ISBN 978-80-248-2399-7.

About Authors

Jiri VYCHODIL, in 2007 finished bachelor study in VSB-Technical University of Ostrava, branch Telecommunication Technologies. Master study finished in 2009, branch Mobile Technologies. Currently works as doctoral student in the Department of Telecommunication at the University.

Martin MIKULEC received the M.Sc. degree from VSB-Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science in 2011. Currently, he is working toward the Ph.D. degree at the Department of Telecommunications. His research is focused on advanced services in IP telephony.

Jakub SAFARIK received his M.Sc. degree in telecommunications from VSB-Technical University of Ostrava, Czech Republic, in 2011 and he continues in studying Ph.D. degree. His research is focused on IP telephony, computer networks and network security.

Miroslav VOZNAK is an Associate Professor with Department of Telecommunications, VSB-Technical University of Ostrava, Czech Republic. He received his M.Sc. and Ph.D. degrees in telecommunications. Topics of his research interests are Next Generation Networks, IP telephony, speech quality and network security.